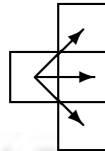


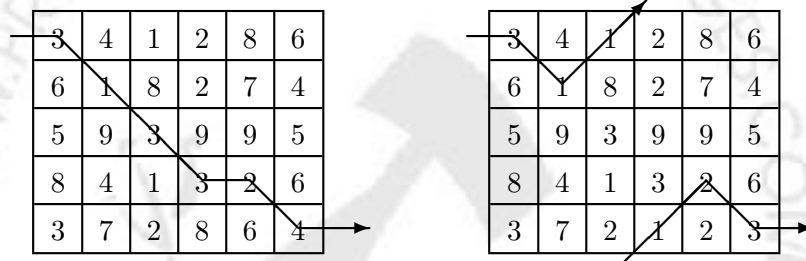
## 111104 Unidirectional TSP

Given an  $m \times n$  matrix of integers, you are to write a program that computes a path of minimal weight from left to right across the matrix. A path starts anywhere in column 1 and consists of a sequence of steps terminating in column  $n$ . Each step consists of traveling from column  $i$  to column  $i + 1$  in an adjacent (horizontal or diagonal) row. The first and last rows (rows 1 and  $m$ ) of a matrix are considered adjacent; i.e., the matrix “wraps” so that it represents a horizontal cylinder. Legal steps are illustrated below.



The *weight* of a path is the sum of the integers in each of the  $n$  cells of the matrix that are visited.

The minimum paths through two slightly different  $5 \times 6$  matrices are shown below. The matrix values differ only in the bottom row. The path for the matrix on the right takes advantage of the adjacency between the first and last rows.



### Input

The input consists of a sequence of matrix specifications. Each matrix consists of the row and column dimensions on a line, denoted  $m$  and  $n$ , respectively. This is followed by  $m \cdot n$  integers, appearing in row major order; i.e., the first  $n$  integers constitute the first row of the matrix, the second  $n$  integers constitute the second row, and so on. The integers on a line will be separated from other integers by one or more spaces. Note: integers are *not* restricted to being positive. There will be one or more matrix specifications in an input file. Input is terminated by end-of-file.

For each specification the number of rows will be between 1 and 10 inclusive; the number of columns will be between 1 and 100 inclusive. No path's weight will exceed integer values representable using 30 bits.

### Output

Two lines should be output for each matrix specification. The first line represents a minimal-weight path, and the second line is the cost of this minimal path. The path consists of a sequence of  $n$  integers (separated by one or more spaces) representing the rows that constitute the minimal path. If there is more than one path of minimal weight, the lexicographically smallest path should be output.

### Sample Input

```
5 6
3 4 1 2 8 6
6 1 8 2 7 4
5 9 3 9 9 5
```

```
8 4 1 3 2 6
3 7 2 8 6 4
5 6
3 4 1 2 8 6
6 1 8 2 7 4
5 9 3 9 9 5
8 4 1 3 2 6
3 7 2 1 2 3
2 2
9 10 9 10
```

### Sample Output

```
1 2 3 4 4 5
16
1 2 1 5 4 5
11
1 1
19
```

